CLAIMS:

1. An auction method for holding an auction for a product comprising the steps of:

receiving bids, for each product type in a transaction, that include minimum desired volumes and maximum desired volumes and evaluation prices for said product;

generating a finite set of bids that include as an element said bids that were received; and

employing dynamic programming to generate, using said bid set, a subset of bids wherein the maximum gain is obtained within a range represented by the count of said product available for sale.

2. The auction method according to claim 1, wherein said evaluation prices for said product are represented as a non-linear function relative to the desired volume of said product type in said transaction.

3. The auction method according to claim 1, further comprising the steps of:

allocating a two-dimensional array V to a memory area by using said dynamic programming;

initializing said two-dimensional array V; and

recursively solving the recursive equation for said two-dimensional array V,

wherein

$$V(k, j) := \max\left\{V(k+1, j),\ V(k, j+1),\ \max_{l_k \le x \le h_k}\{V(k+1, j+x) + e_k(x)\}\right\}$$

is used as the recursive equation, where k denotes an

integer equal to or greater than 1 and equal to or smaller than n; j denotes an integer equal to or greater than 0 and equal to or smaller than s; n denotes the number of bids; s denotes the number of products available for the transaction; e_k denotes the evaluation price when x units of products are purchased according to the bid b_k; l_k denotes the minimum volume of the bid b_k; and h_k denotes the maximum volume of the bid b_k.

4. The auction method according to claim 3, wherein a bid according to which said product is optimally distributed is selected by back tracking of said two-dimensional array V from the element on the smallest row and in the smallest column.

5. The auction method according to claim 1, further comprising:

allocating two-dimensional arrays V and Q to a memory area by using said dynamic programming;

initializing said two-dimensional arrays V and Q; and

recursively solving recursive equations for said two-dimensional arrays V and Q,

wherein said evaluation prices for said product represent a linear function relative to the volumes for said product desired for said transaction, and

wherein

$$V(k,j):=\max\left\{\begin{array}{ll} V(k+1,j) & \\ V(k,j+1) & \\ V(k,j+1)+e_k & (if\ l_k \le Q(k,j+1) < h_k) \\ V(k+1,j+l_k)+e_k l_k & \end{array}\right\}$$

$$Q(k,j):=\left\{\begin{array}{ll} Q(k,j+1)+1 & (if\ V(k,j)=V(k,j+1)+e_k) \\ l_k & (if\ V(k,j)=V(k+1,j+l_k)+e_k l_k) \\ Q(k,j+1) & (if\ V(k,j)=V(k,j+1)) \\ 0 & (otherwise) \end{array}\right\}$$

is employed as said recursive equation, where k denotes an integer equal to or greater than 1 and equal to or smaller than n; j denotes an integer equal to or greater than 0 and equal to or smaller than s; n denotes the number of bids; s denotes the number of products available for the transaction; e_k denotes the evaluation price in the bid b_k for a product for each unit; l_k denotes the minimum volume of the bid b_k; and h_k denotes the maximum volume of the bid b_k.

6. The auction method according to claim 5, wherein a bid according to which said product is optimally distributed is selected by back tracking of said two-dimensional array V from the element on the smallest row and in the smallest column.

7. An auction method for performing an auction for multiple products of multiple types comprising the steps of:

receiving bids that each include a combination of said products (including only one type of one product), the volume of said combination desired for the transaction and an evaluation price for said combination;

generating a finite set of bids that include said bids

as elements; and

employing dynamic programming to generate, using said bid set, a subset of bids wherein a maximum gain is obtained within a range represented by the count of said individual products that are available for sale,

wherein said combination of said products satisfies either a first condition $C_i \cap C_j \neq \phi$ or a second condition $C_i \subset C_j$, wherein C_i and C_j $(i<j)$ are two different arbitrary combinations.

8. The auction method according to claim 7, wherein the minimum volume and the maximum volume for said combination desired for the transaction are designated as said volumes for said combination.

9. The auction method according to claim 7, wherein said evaluation prices for said combination are represented as a non-linear function relative to the volume of said combination desired for the transaction.

10. The auction method according to claim 7, further comprising:

a step of sorting said bids in accordance with each combination type by using said dynamic programming method;

a step of allocating a two-dimensional array V for said sorted combination C_i;

an initialization step, which includes the steps of

determining whether a child set of said combination C_i is an empty set and substituting, when the

determination is true, 0 into each element on the (n+1)th row of said two-dimensional array V, and

substituting, when the determination is false, the sum of the elements on the first row of said two-dimensional array V, for all the child sets of said combination C_i, into all the elements on the (n+1)th row of said two-dimensional array V; and

a step of recursively solving said recursive equation of said two-dimensional array V.

Claim 11

The auction method according to claim 10, wherein, as said recursive equation employed is either

$V(k,j):=\max\{V(k+1,j),V(k,j+1),V(k+1,j+r_k)+r_k e_k\}$

where k denotes an integer equal to or greater than 1 and equal to or smaller than n; j denotes an integer equal to or greater than 0 and equal to or smaller than s; n denotes the number of bids for said combination C_i; s denotes the minimum number of products available for the transaction, which is included in said combination C_i; e_k denotes the evaluation price in the bid b_k for one combination; and r_k denotes the volumes of the combinations for the bid b_k, or

$V(k,j):=\max\{V(k+1,j),V(k,j+1),\max_{l_k \le x \le h_k}\{V(k+1,j+x)+e_k(x)\}\}$

where k denotes an integer equal to or greater than 1 and equal to or smaller than n; j denotes an integer equal to or greater than 0 and equal to or smaller than s; n denotes the number of bids for said combination C_i; s denotes the minimum number of products available for the transaction, which is included in said combination C_i; e_k denotes the evaluation

price when x units of products are purchased according to the bid b_k; l_k denotes the minimum volume of the bid b_k; and h_k denotes the maximum volume of the bid b_k, or

$$V(k, j) := \max\left\{\begin{array}{ll} V(k+1, j) & \\ V(k, j+1) & \\ V(k, j+1) + e_k & (\text{if } l_k \leq Q(k, j+1) < h_k) \\ V(k+1, j+l_k) + e_k l_k & \end{array}\right\}$$

$$Q(k, j) := \left\{\begin{array}{ll} Q(k, j+1) + 1 & (\text{if } V(k, j) = V(k, j+1) + e_k) \\ l_k & (\text{if } V(k, j) = V(k+1, j+l_k) + e_k l_k) \\ Q(k, j+1) & (\text{if } V(k, j) = V(k, j+1)) \\ 0 & (\text{otherwise}) \end{array}\right\}$$

where k denotes an integer equal to or greater than 1 and equal to or smaller than n; j denotes an integer equal to or greater than 0 and equal to or smaller than s; n denotes the number of bids for said combination C_i; s denotes the minimum number of products available for the transaction, which is included in said combination C_i; e_k denotes the evaluation price in the bid b_k for a product for one combination; l_k denotes the minimum volume of the bid b_k; and h_k denotes the maximum volume of the bid b_k.

12. The auction method according to claim 11, wherein, for a set C^R that is a subset of the whole set C and has said combination C_i as one element, but whose other elements are not child sets of the other elements of said set C, said two-dimensional array V is tracked backward, beginning with the element on the minimum row and in the minimum column, and after the element on the (n+1)th row is reached, from the first row of a child set of said element on the (n+1)th row, said two-dimensional array V is tracked further

backward to select a bid for the optimal distribution of said products.

13. An auction system for holding an auction for a product comprising:

means for receiving bids, for each product type in a transaction, that include minimum desired volumes and maximum desired volumes and evaluation prices for said product;

means for generating a finite set of bids that include as an element said bids that were received; and

means for employing dynamic programming to generate, using said bid set, a subset of bids wherein the maximum gain is obtained within a range represented by the count of said product available for sale.

14. The auction system according to claim 13, wherein said evaluation prices for said product are represented as a non-linear function relative to the desired volume of said product type in said transaction.

15. The auction system according to claim 13, further comprising:

means for allocating a two-dimensional array V to a memory area by using said dynamic programming;

means for initializing said two-dimensional array V; and

recursively solving the recursive equation for said two-dimensional array V,

wherein

$$V(k, j) := \max\{V(k+1, j), V(k, j+1), \max_{l_k \le x \le h_k}\{V(k+1, j+x) + e_k(x)\}\}$$

is used as the recursive equation, where k denotes an integer equal to or greater than 1 and equal to or smaller than n; j denotes an integer equal to or greater than 0 and equal to or smaller than s; n denotes the number of bids; s denotes the number of products available for the transaction; e_k denotes the evaluation price when x units of products are purchased according to the bid b_k; l_k denotes the minimum volume of the bid b_k; and h_k denotes the maximum volume of the bid b_k.

16. The auction system according to claim 15, further comprising:

means for selecting a bid according to which said product is optimally distributed by back tracking of said two-dimensional array V from the element on the smallest row and in the smallest column.

17. The auction system according to claim 13, further comprising:

means for allocating two-dimensional arrays V and Q to a memory area by using said dynamic programming;

means for initializing said two-dimensional arrays V and Q; and

means for recursively solving recursive equations for said two-dimensional arrays V and Q,

wherein said evaluation prices for said product represent a linear function relative to the volumes for said

product desired for said transaction, and

wherein

$$V(k,j):=\max\left\{\begin{array}{ll} V(k+1,j) & \\ V(k,j+1) & \\ V(k,j+1)+e_k & (if\ l_k \le Q(k,j+1) < h_k) \\ V(k+1,j+l_k)+e_k l_k & \end{array}\right\}$$

$$Q(k,j):=\left\{\begin{array}{ll} Q(k,j+1)+1 & (if\ V(k,j)=V(k,j+1)+e_k) \\ l_k & (if\ V(k,j)=V(k+1,j+l_k)+e_k l_k) \\ Q(k,j+1) & (if\ V(k,j)=V(k,j+1)) \\ 0 & (otherwise) \end{array}\right\}$$

is employed as said recursive equation, where k denotes an integer equal to or greater than 1 and equal to or smaller than n; j denotes an integer equal to or greater than 0 and equal to or smaller than s; n denotes the number of bids; s denotes the number of products available for the transaction; e_k denotes the evaluation price in the bid b_k for a product for each unit; l_k denotes the minimum volume of the bid b_k; and h_k denotes the maximum volume of the bid b_k.

18. The auction system according to claim 17, wherein a bid according to which said product is optimally distributed is selected by back tracking of said two-dimensional array V from the element on the smallest row and in the smallest column.

19. An auction system for performing an auction for multiple products of multiple types comprising:

means for receiving bids that each include a combination of said products (including only one type of one product), the volume of said combination desired for the

transaction and an evaluation price for said combination;

means for generating a finite set of bids that include said bids as elements; and

means for employing dynamic programming to generate, using said bid set, a subset of bids wherein a maximum gain is obtained within a range represented by the count of said individual products that are available for sale,

wherein said combination of said products satisfies either a first condition $C_i \cap C_j \neq \phi$ or a second condition $C_i \subset C_j$, wherein C_i and C_j $(i<j)$ are two different arbitrary combinations.

20. The auction system according to claim 19, wherein the minimum volume and the maximum volume for said combination desired for the transaction are designated as said volumes for said combination.

21. The auction system according to claim 19, wherein said evaluation prices for said combination are represented as a non-linear function relative to the volume of said combination desired for the transaction.

22. The auction system according to claim 19, further comprising:

means for sorting said bids in accordance with each combination type by using said dynamic programming method;

means for allocating a two-dimensional array V for said sorted combination C_i;

initialization means, which includes

means for determining whether a child set of said combination C_i is an empty set and substituting, when the determination is true, 0 into each element on the (n+1)th row of said two-dimensional array V, and

means for substituting, when the determination is false, the sum of the elements on the first row of said two-dimensional array V, for all the child sets of said combination C_i, into all the elements on the (n+1)th row of said two-dimensional array V; and

means for recursively solving said recursive equation of said two-dimensional array V.

23. The auction system according to claim 22, wherein, as said recursive equation employed is either

$V(k,j):=\max\{V(k+1,j), V(k,j+1), V(k+1,j+r_k)+r_k e_k\}$

where k denotes an integer equal to or greater than 1 and equal to or smaller than n; j denotes an integer equal to or greater than 0 and equal to or smaller than s; n denotes the number of bids for said combination C_i; s denotes the minimum number of products available for the transaction, which is included in said combination C_i; e_k denotes the evaluation price in the bid b_k for one combination; and r_k denotes the volumes of the combinations for the bid b_k, or

$V(k, j):=\max\left\{V(k+1, j), V(k, j+1), \max_{1_k \le x \le h_k}\{V(k+1, j+x)+e_k(x)\}\right\}$

where k denotes an integer equal to or greater than 1 and equal to or smaller than n; j denotes an integer equal to or greater than 0 and equal to or smaller than s; n denotes the number of bids for said combination C_i; s denotes the minimum number of products available for the transaction, which is

included in said combination C_i; e_k denotes the evaluation price when x units of combinations are purchased according to the bid b_k; l_k denotes the minimum volume of the bid b_k; and h_k denotes the maximum volume of the bid b_k, or

$$V(k,j) := \max\left\{\begin{array}{ll} V(k+1,j) & \\ V(k,j+1) & \\ V(k,j+1)+e_k & (if\ l_k \le Q(k,j+1) < h_k) \\ V(k+1,j+l_k)+e_k l_k & \end{array}\right\}$$

$$Q(k,j) := \left\{\begin{array}{ll} Q(k,j+1)+1 & (if\ V(k,j)=V(k,j+1)+e_k) \\ l_k & (if\ V(k,j)=V(k+1,j+l_k)+e_k l_k) \\ Q(k,j+1) & (if\ V(k,j)=V(k,j+1)) \\ 0 & (otherwise) \end{array}\right\}$$

where k denotes an integer equal to or greater than 1 and equal to or smaller than n; j denotes an integer equal to or greater than 0 and equal to or smaller than s; n denotes the number of bids for said combination C_i; s denotes the minimum number of products available for the transaction, which is included in said combination C_i; e_k denotes the evaluation price in the bid b_k for a product for one combination; l_k denotes the minimum volume of the bid b_k; and h_k denotes the maximum volume of the bid b_k.

24. The auction system according to claim 23, wherein, for a set C^R that is a subset of the whole set C and has said combination C_i as one element, but whose other elements are not child sets of the other elements of said set C, said two-dimensional array V is tracked backward, beginning with the element on the minimum row and in the minimum column, and after the element on the (n+1)th row is reached, from the first row of a child set of said element on the (n+1)th

row, said two-dimensional array is tracked further backward to select a bid for the optimal distribution of said products.

25. A computer-readable storage medium on which a program for holding an auction for a product is stored, said program permitting a computer to perform:

a function for receiving bids, for each product type in a transaction, that include minimum desired volumes and maximum desired volumes and evaluation prices for said product;

a function for generating a finite set of bids that include as an element said bids that were received; and

a function for employing dynamic programming to generate, using said bid set, a subset of bids wherein the maximum gain is obtained within a range represented by the count of said product available for sale.

26. A computer-readable storage medium on which a program is stored for performing an auction for multiple products of multiple types, said program permitting a computer to perform:

a function for receiving bids that each include a combination of said products (including only one type of one product), the volume of said combination desired for the transaction and an evaluation price for said combination;

a function for generating a finite set of bids that include said bids as elements; and

a function for employing dynamic programming to

generate, using said bid set, a subset of bids wherein a maximum gain is obtained within a range represented by the count of said individual products that are available for sale,

wherein said combination of said products satisfies either a first condition $C_i \cap C_j \neq \phi$ or a second condition $C_i \subset C_j$, wherein C_i and C_j $(i<j)$ are two different arbitrary combinations.

27. An auction method for holding an auction for a product comprising the steps of:

receiving bids, for each product type in a transaction, that include a condition concerning said product;

generating a finite set of bids that include as an element said bids that were received; and

employing dynamic programming to generate, using said bid set, a subset of bids wherein the maximum gain is obtained within a range represented by the count of said product available for sale.